

METHOD AND APPARATUS FOR ACCESSING INSTRUMENTATION DATA FROM
WITHIN A MANAGED CODE ENVIRONMENT

Application Serial No.: 09/900,060

Attorney Docket No.: 13768.1180.1

Examiner: Tuan Vu

Listing of Claims:

1. (Currently Amended) A computer-implemented method for providing access to instrumentation data from within a managed code runtime environment, wherein the data arises from a source outside the managed code runtime environment, the method comprising:

providing an application compiled into an intermediate form from a runtime-aware programming language, the application being suitable for execution by a runtime engine in a managed code runtime environment;

executing the application in the managed code runtime environment;

receiving a request at the runtime engine from the executing application for instrumentation data from a source executing in a native code environment outside the managed code runtime environment;

the runtime engine binding an instance of a management object class to a particular management instrumentation object, the instance of the management object class being constructed from an API comprising the management object class, the management object class encapsulating a method for binding an instance of the management object to the particular management instrumentation object and a method for saving changes made to the object,

the API further comprising a management object searcher class which permits the retrieval of a collection of instrumentation objects based on a specified query, a management event watcher class, a management path class, a management scope class providing functionality for building a management scope or namespace, and a management options class which is a base class for subclasses that allow options to be set that are utilized when certain operations are performed, the certain operations comprising one or more of connecting to a management scope, enumeration of objects, and querying,

the particular management instrumentation object representing the data source executing in the native code environment outside the managed code environment, {H-H}

wherein binding the instance comprising comprises passing a path of an instrumentation data object for the instance of the management object to bind to, for accessing the

~~instrumentation data~~, options used to retrieve the instrumentation data object, and an ~~identification of a namespace identifying~~ a parent of the instrumentation data object;

transmitting a corresponding request for said instrumentation data to the source executing in the native code environment outside the managed code runtime environment;

receiving a response to said corresponding request from said instrumentation data source;

converting said response to a format that is compatible with said managed code runtime environment; and

responding to said request for instrumentation data with said converted response.

2. (Canceled)

3. (Currently Amended) The method of Claim 1, wherein converting said response comprises converting the instrumentation data object to a management object that is compatible with said managed code runtime environment.

4. (Cancelled)

5. (Previously Presented) The method of Claim 3, wherein said response comprises an indication that an operation was unsuccessful and wherein converting said response to said format comprises generating a management exception object.

6. (Currently Amended) The method of Claim 5, wherein said indication that an operation was unsuccessful comprises at least one error code, ~~{+}~~

7. (Currently Amended) A computer-readable storage medium comprising instructions which, when executed by a computer, cause the computer to perform the method of claim 1, ~~any one of Claims 1 and 3-6~~.

8. (Currently Amended) A computer-controlled apparatus comprising a processing unit and a system memory, and wherein the apparatus further comprises a managed code runtime environment and is configured to carry out the method of claim 1, ~~any one of Claims 1 and 3-6.~~

9. (Currently Amended) A computer-implemented method for accessing instrumentation data from within a managed code runtime environment, wherein the managed code runtime environment provides a runtime engine that executes an application compiled from source written in a runtime-aware language into an intermediate form, the method comprising:

receiving a request from an application compiled from source written in a runtime-aware language into an intermediate form for instrumentation data representing management information about other applications and devices available in an environment outside the managed code runtime environment, the outside environment comprising a native code environment,

~~the request comprising a path of an instrumentation data object for accessing said instrumentation data, options used to retrieve the instrumentation data object, and a namespace of the instrumentation data object;~~

~~in response to said request, querying for said instrumentation data using the path of said instrumentation data object for accessing said instrumentation data;~~

~~determining whether said instrumentation data was successfully returned;~~

~~in response to determining that said instrumentation data was successfully returned, constructing a management object in the runtime environment;~~

wherein the request for said instrumentation data from said outside environment is implemented within a runtime engine of said managed code runtime environment using an API for binding an instance of a management object class to a particular instrumentation data object, the instance of the management object class being constructed from the API and encapsulating a method for binding an instance of the management object to the particular instrumentation data object and a method for saving changes made to the object, the instrumentation data object representing the data source executing in the native code environment outside the managed code runtime environment;

wherein binding the management class instance to the instrumentation data object comprises passing the path of the instrumentation data object to bind to, the options used to retrieve the instrumentation data object, and a namespace identifying the parent of the instrumentation data object;

the management object being constructed from an wherein the API which comprises a management object class{{}} which encapsulates a method for binding an instance of the

~~management object to the instrument object and a method for saving changes made to the object, a management object searcher class which permits the retrieval of a collection of instrumentation objects based on a specified query, a management event watcher class, a management path class, a management scope class providing functionality for building a management scope or namespace, a management exception class, and a management options class which is a base class for subclasses that allow options to be set that are utilized when certain operations are performed, the certain operations comprising one or more of connecting to a management scope, enumeration of objects, and querying; {and}}~~
~~populating the management object with said instrumentation data~~
~~converting a response to a format that is compatible with the managed code runtime environment; and~~
~~responding to the request for instrumentation data with the converted response.~~

10. (Currently Amended) The method of Claim 9, further comprising: ~~wherein constructing said management object in the runtime environment and populating said management object with said instrumentation data includes~~ binding ~~{{and}}~~ the instance of ~~{{the}}~~ the management object class to said instrumentation data object for accessing said instrumentation data.

11. (Cancelled)

12. (Previously Presented) The method of Claim 10, further comprises constructing a management path object identifying the path to said instrumentation data object.

13. (Currently Amended) The method of Claim 10, further comprising constructing a management options object specifying ~~{{the}}~~ one or more options to retrieve said instrumentation data object for accessing said instrumentation data.

14. (Original) The method of Claim 10, further comprising:
throwing a management exception object in response to determining that said instrumentation data was not successfully returned.

15. (Previously Presented) The method of Claim 10, wherein properties of said management object may be accessed utilizing an indexer.

16. (Currently Amended) A computer-readable storage medium comprising instructions which, when executed by a computer, cause the computer to perform the method of claim 9, any one of Claims 9-15.

17. (Currently Amended) A computer-controlled apparatus comprising a processing unit and a system memory, and wherein the apparatus further includes a managed code runtime environment and is configured to carry out the method of ~~claim 9, any one of Claims 9-13,~~

August 11, 2009

Thomas M. Bonacci
Reg. No. 63,368

WORKMAN | NYDEGGER

1000 Eagle Gate Plaza
60 East South Temple
Salt Lake City, Utah 84111
Telephone (801) 533-9800
Facsimile (801) 328-1707
tbonacci@wnlaw.com